

APPLICATION FOR UNITED STATES LETTERS PATENT

For

**INVOICE DATA OBJECT FOR A COMMON DATA OBJECT FORMAT**

Inventors:

Nardo Catahan  
Ying-Chieh Lan  
Roy Liu  
Joshua Roper

Prepared by:

BLAKELY SOKOLOFF TAYLOR & ZAFMAN LLP  
32400 Wilshire Boulevard  
Los Angeles, CA 90025-1026  
(408) 720-8598

Attorney's Docket No.: 005306.P104

“Express Mail” mailing label number: EV339922851US

## **INVOICE DATA OBJECT FOR A COMMON DATA OBJECT FORMAT**

### **CLAIM OF PRIORITY**

[0001] This application is related to, and hereby claims the benefit of provisional application number 60/451,984 which was filed March 4, 2003.

### **FIELD**

[0002] Embodiments of the invention relate generally to computer software application, and more specifically to common data object formats for such applications.

### **BACKGROUND**

[0003] Various business entities, such as companies, store information electronically in furtherance of their business needs. These companies may have extensive databases of information that include customer tables, supplier tables, employee tables, and so on. The structure of the database system (schemas) and the data object format (DOF) of each database may be customized to help meet the business needs of the company. For example, an automotive manufacturer may organize information about its customers in a way that is very different from the way that an online bookstore may organize information about its customers. Even within a single company, that company may use many different application programs that employ very different schemas and DOFs. For example, a customer relationship management application program may use a DOF that is very different from the DOF used by an accounting program. The use of customized DOFs by a company and by applications within the company has the advantage that it allows information to be modeled in a way that is appropriate for the business needs of the company. Unfortunately, because of this diversity in the DOFs, it is not easy for the

company to share its information with other companies or for applications to share their information.

[0004] The inter-exchange of information between applications of different business entities or even between different applications of the same business entity can be problematic due to the variation in DOFs between applications.

[0005] For example, a business entity may use a proprietary billing system. If the business entity decides to integrate a number of related applications from each of several software vendors, a translation mechanism may have to be created and implemented between the underlying billing system and each related application. This is because each application from a different software vendor may have a unique, or substantially different, DOF. Moreover, full integration of the multiple applications may require creation and implementation of a translation mechanism between each of the related applications as well.

[0006] A change in the underlying billing system may necessitate recreating and implementing such translation mechanisms.

[0007] Various attempts have been made to define standard data models so that information can be more easily shared between companies and applications. For example, the Open Applications Group has designed a standard data model that can be used by companies and applications when sharing information. A problem with such data models is that they did not provide effective ways to model relationships between various parties, such as a person or a company. In addition, if a company or an application developer wants to customize the standard data model, the customized data model may not be compatible with future upgrades of the standard data model. It would be desirable to have a data model that would more effectively model relationships and facilitate the upgrading of customizations of the data model.

## **BRIEF DESCRIPTION OF THE DRAWINGS**

[0008] **Figure 1** illustrates a process by which a common DOF for invoice information is implemented to effect the inter-exchange of invoice information between business applications employing disparate DOFs for invoice information in accordance with one embodiment of the invention;

[0009] **Figure 2** illustrates the interconnection between a plurality of various business system applications and a universal business application network to effect the inter-exchange of invoice information between the business applications in accordance with one embodiment of the invention;

[0010] **Figure 3** illustrates an exemplary architecture for a universal business application network in accordance with one embodiment of the invention;

[0011] **Figures 4A – 4G** illustrate an exemplary data structure for a common DOF in accordance with one embodiment of the invention; and

[0012] **Figure 5** illustrates a process by which custom data is added to an invoice class in accordance with one embodiment of the invention.

## **DETAILED DESCRIPTION**

### **OVERVIEW**

[0013] Embodiments of the invention provide methods and data structures for the effective and efficient synchronization or inter-exchange of invoice information between business applications employing disparate DOFs. For one embodiment, a DOF is provided that allows for relationships between entities, also referred to as invoices, to be modeled as attributes of an entity and for customization of the DOF in a manner that facilitates upgrading of the DOF. For one embodiment, the invoice DOF is provided in a common software language (i.e., software specification). In one embodiment, the common DOF defines an invoice class that includes multiple data types and the relationships between the data types of the invoice class. The relationships may include basic elements of invoice DOFs from various business applications.

[0014] For one embodiment, a method is provided for efficient synchronization or inter-exchange of invoice information between business applications using different invoice DOFs. For such an embodiment, invoice information from each of several business applications is translated to a common DOF. The invoice information, in the common DOF, is then inter-exchanged among the several business applications. Each application has only to translate the invoice information from the common DOF to the application-specific DOF of the respective business application.

[0015] In the following description, numerous specific details are set forth. However, it is understood that embodiments of the invention may be practiced without these specific details. In other instances, well-known circuits, structures and techniques have not been shown in detail in order not to obscure the understanding of this description.

[0016] Reference throughout the specification to “one embodiment” or “an embodiment” means that a particular feature, structure, or characteristic described in connection with the embodiment is included in at least one embodiment of the present invention. Thus, the appearance of the phrases “in one embodiment” or “in an embodiment” in various places throughout the specification are not necessarily all referring to the same embodiment. Furthermore, the particular features, structures, or characteristics may be combined in any suitable manner in one or more embodiments.

[0017] Moreover, inventive aspects lie in less than all features of a single disclosed embodiment. Thus, the claims following the Detailed Description are hereby expressly incorporated into this Detailed Description, with each claim standing on its own as a separate embodiment of this invention.

## **PROCESS**

[0018] **Figure 1** illustrates a process by which a common DOF for invoice information is implemented to effect the inter-exchange of invoice information between business applications employing disparate DOFs for invoice information in accordance with one embodiment of the invention. Process 100, shown in Figure 1, begins at operation 105 in which a base set of essential elements to describe invoice information is determined. For example, for one embodiment the essential elements may be determined to include a common identification object, invoice base data, pricing data, shipping data, and list of invoice line item details consisting of all the detail information of an invoice. For one embodiment, base data contains the base information of an invoice, which is fundamental data sufficient to identify the invoice. Base data may contain the invoice identification information and other fundamental data.

[0019] For one embodiment, essential elements may be determined so as to achieve a specified level of compatibility with the DOFs of various extant business applications.

[0020] At operation 110 a common DOF for the invoice information is created. For one embodiment, the common DOF includes the determined essential elements. For various alternative embodiments, the common DOF may include some or all of the determined essential elements as well as other elements. The common DOF is created in a common format that may be selected based upon the extent to which the format is interoperable with various business applications. For one embodiment, the common DOF is created in extensible markup language (XML) format that allows application designers to create customized tags that enable the transmission, validation, and interpretation of data between applications.

[0021] At operation 115 the invoice information from a plurality of business applications having different invoice DOFs is translated into the common DOF. That is, for each application, the invoice information in an application-specific DOF is translated into the common DOF.

[0022] At operation 120 the invoice information in the common DOF is exchanged between two or more of the business applications. At this point a business integration server completes the translation of the invoice information in the common DOF to the application-specific DOF for each respective business application as described below.

#### **SYSTEM**

[0023] **Figure 2** illustrates the interconnection between a plurality of various business system applications and a universal business application network to effect the inter-exchange of invoice information between the business applications in accordance with one embodiment of the invention. System 200, shown in Figure 2, includes a number of business systems 202, each having an application using an application-specific DOF for invoice information. The business

systems are coupled through a universal business application network 201 that serves as an integration hub for the business systems.

[0024] In accordance with one embodiment of the invention, each of the business systems implements a translation mechanism to translate invoice information, in an application-specific DOF, into a common DOF. The invoice information in the common DOF may then be inter-exchanged between the business systems through the universal business application network. A business integration server then translates the invoice information from the common DOF into a particular application-specific DOF for a respective business system as described more fully below in reference to Figure 3.

[0025] The architecture of the universal business application network allows new business applications that access legacy business systems to be developed with minimum customization. The legacy business systems can be provided by a single business organization or by different business organizations. The universal business application network also allows the business applications to exchange invoice information using an invoice common DOF. In one embodiment, the universal business application network uses the XML and Web services standards.

[0026] **Figure 3** illustrates an exemplary architecture for a universal business application network in accordance with one embodiment of the invention. The hub of the universal business application network 300 is the business integration server 310 that connects to the various business systems 301 via adapters 302. The business integration server includes a transport layer 311, an object model 312, a transformation store 313, a business process controller 314, and a business process store 315. The transport layer 311 is a mechanism through which business information is exchanged between the business systems 301 and the business integration server



310. Each business system 301 may have an adapter 302 that is appropriate to the protocol of the transport layer 311. For example, the transport mechanism may use communications protocols such as TCP/IP. The transport layer may provide a messaging service for queuing, for guaranteeing delivery of messages, and for handling both synchronous and asynchronous messaging. The adapters 302 relay events from the business systems 301 to the business integration server 310 and can import configurations of the business systems 301 into the business integration server 310. In addition, the universal business application network 300 may include encryption and authentication mechanisms to ensure the security and integrity of the information. For example, authentication will help ensure that a business process is accessing the intended business system, rather than an impostor business system.

[0027] As discussed above, the common DOF may include the definition of various invoice-related objects. The objects may be defined using standard object definition tools such as an XML schema definition tool. The transformation store contains transformations for translating information received from the business systems to the common DOF, and vice versa. For example, an invoice object may include a globally unique identifier for each person. A transformation for a business system that does not use globally unique identifiers may need to access an identification server to determine the globally unique identifier for each invoice. The transformations may be specified as a computer program, an XML Stylesheet Language Transform ("XSL T"), etc. The business process store contains the business processes that have been defined. A business process may be specified as a script, a process flow, an executable program, etc. In one embodiment, the business processes are defined using the Web Services Flow Language ("WSFL"). The business processes orchestrate a sequence of steps across multiple applications provided by the business systems to achieve a business objective. The

business process controller coordinates the execution of the business processes. The business process controller may instantiate objects and invoke functions of the objects in accordance with the various business processes. The business process controller may also initiate the execution of business processes based on predefined conditions and events. For example, the business process controller may launch a certain business process each time an alert is received. Although not shown, the business integration network may provide a standard library of business routines that may be invoked by the business processes. For example, a standard business routine might be to identify whether two invoice objects represent the same individual or to apply business rules to various objects and take the appropriate action as defined by those rules. The business integration server may also include various tools to facilitate the development of business processes. These tools may aid in the development of transformations, the defining of common objects, and the writing of process flows.

#### **DATA STRUCTURE**

[0028] The common DOF may include basic elements of invoice DOFs from various business applications. For example, the common DOF may include a common identification object, invoice base data, pricing data, shipping data, and list of invoice line item details among its base element set. Additionally, for alternative embodiments, the common DOF may include such elements as related business unit, list of related parties, related invoice type, related invoice plan, related inventory location, related price list, related payment terms, list of payment methods, list of payments, and list of comments.

[0029] In one embodiment, the common DOF defines a hierarchy of the data elements for describing an invoice. The common DOF may define data elements that are complex. A complex data element is a data element that comprises data sub-elements. For example, a

payment method element may be a complex data element that includes card payment, cash payment, and finance account payment data sub-elements.

[0030] **Figures 4A – 4G** illustrate an exemplary data structure for a common DOF in accordance with one embodiment of the invention. One skilled in the art will appreciate that the name of each data element is descriptive of the information stored in the data element.

[0031] **Figure 4A** illustrates the highest level data elements 401 of the invoice class in accordance with one embodiment. The highest level data elements include id, baseData, pricingData, shippingData, relatedBusUnit, listOfRelatedParty, relatedInvoiceType, relatedInvoicePlan, relatedInventoryLocation, relatedPriceList, relatedPaymentTerm, listofPaymentMethod, listOfRelatedPayment, listOfComment, listOfInvoiceLine, and customData data elements.

[0032] The id data element may be a unique identifier of a party. The customData data element initially contains no data elements, but custom data elements can be added by defining data elements in the CustomDataType as described below.

[0033] **Figure 4B** illustrates the data elements of the Related Party class 402 in accordance with one embodiment. The Related Party class represents the related party information. The Related Party class includes id, communicationData, dataCleansingData, listOfAddress, listOfRelationship, listOfAlternateId, listOfLicenseData, customPartyData, baseData, and customData. The Related Party class also includes a customData data element with a type of CustomDataType that initially is defined to have no data elements.

[0034] **Figure 4C** illustrates the data elements of the Payment Method class 403 in accordance with one embodiment. The Payment Method class includes cardPayment, cashPayment, financeAccountPayment, purchaseOrderPayment and customData data elements.

[0035] **Figure 4D** illustrates the data elements of the Related Payment class 404 in accordance with one embodiment. The Related Payment class includes id, baseData, paymentData, listOfRelatedInvoice, PaymentMethod, listOfComment, and customData data elements.

[0036] **Figure 4E** illustrates the data elements of the Comment class 405 in one embodiment. The Comment class includes textCode and text data elements.

[0037] **Figure 4F** illustrates the data elements of the Invoice Line class 406 in accordance with one embodiment. The Invoice Line represents the related invoice line item detail information for an invoice. The Invoice Line class includes id, baseData, pricingData, shippingData, relatedProduct, relatedInstalledProduct, listOfRelatedParty, relatedInventoryLocation, relatedPriceList, relatedPaymentTerm, relatedInvoicePlan, relatedOrderLine, relatedShipmentLine, listOfComment, and customData data elements.

[0038] **Figure 4G** illustrates data elements for various information related to the invoice information in accordance with one embodiment of the invention. The related business unit class 410 represents the related organization information. The related invoice type class 411 represents the related invoice type information for the respective invoice, such as invoice, credit memo, etc. The related invoice plan class 412 represents the related invoice plan information for the respective invoice, such as 1<sup>st</sup> of the month for 24 months, yearly on January 1<sup>st</sup>, etc. The related inventory location class 413 represents related inventory location. The related price list class 414 represents related price lists. The related payment term class 415 represents related payment terms.

[0039] Embodiments of the invention provide a common DOF for invoice information that can be used as an intermediate DOF during translation of invoice information from one application-specific DOF to another.

[0040] For one embodiment, the common DOF may contain a custom data element at various places within the hierarchy of data elements that allow a customer to put in more attributes. A custom data element is of a custom data element type. The custom data element type initially defines no data elements. The data model can be customized by defining custom data elements for the custom data element type. For example, the data elements relating to the relationship of an invoice may have a custom data element through which data elements relating to the history of previously related invoices can be defined. Because the custom data elements are defined at various places within the hierarchy, the customizations of the data model can be associated with related data elements within the hierarchy.

[0041] In one embodiment, each of the types of an invoice specifies a custom data element for that type. For example, the related party data element may be defined as the related party data type. If so, the data type can be customized by adding data elements to the definition of the related party data type. The definition may be stored in a file that is separate from the file in which the data type is defined. A portion of an XML schema that defines the custom data a related party is

```
<xs:element name="customData" type=
"custom:Related Party Data Type" minOccurs = "0" />
```

where "custom" specifies a file that contains the definition of Related Party Data Type, which may be

```
<xs:complexType name=Related PartyDataType">
<xs:annotation
<xs:documentation>
Define the custom data element for this type following this annotation
<xs:documentation>
</xs:annotation>
</xs:complexType>
```

[0042] **Figure 5** illustrates a process by which custom data is added to an invoice class in accordance with one embodiment of the invention. Process 500, shown in Figure 5, begins at operation 505 in which the schema for the invoice class is retrieved. The schema may be an XML schema file that includes a custom data element of a type that is defined in another file.

[0043] At operation 510 the schema for the types of custom data is retrieved and opened. The schema may be stored in an XML schema file that contains the definition for each type of custom data.

[0044] At operation 515 the tags relating to the custom data type of interest are located and the custom data elements are added to the tags.

[0045] At operation 520, the custom data schema with the newly defined data elements added to the custom data type is closed.

[0046] Embodiments of the invention include various operations. Many of the methods are described in their most basic form, but operations can be added to or deleted from any of the methods without departing from the basic scope of the invention.

[0047] It will be apparent to those skilled in the art that the data structure and operations of embodiments of the invention may be stored upon or embodied in machine-executable instructions, which may be used to cause a general-purpose or special-purpose processor or logic circuits programmed with the instructions to perform specific operations.

[0048] Alternatively, the operations of embodiments of the invention may be performed by a combination of hardware and software. Embodiments of the invention present may be provided as a computer program product that may include a machine-readable medium having stored thereon instructions, which may be used to program a computer (or other electronic devices) to perform a process according to various embodiments of the invention. Likewise, embodiments

of the invention present may be provided as data structures stored upon a machine-readable medium. Such machine-readable medium may include, but are not limited to, floppy diskettes, optical disks, CD-ROMs, and magneto-optical disks, ROMs, RAMs, EPROMs, EEPROMs, magnet or optical cards, flash memory, or other type of media / machine-readable medium suitable for storing electronic instructions. Moreover, the invention may also be downloaded as a computer program product, wherein the program may be transferred from a remote computer to a requesting computer by way of data signals embodied in a carrier wave or other propagation medium via a communication cell (e.g., a modem or network connection).

[0049] The computers (e.g., universal business application network computer and business systems computer) may include a central processing unit, memory, input devices (e.g., keyboard and pointing devices), output devices (e.g., display devices), and storage devices (e.g., disk drives) The memory and storage devices may be computer-readable media that may contain instructions that implement the security system. In addition, the data structures and message structures may be stored or transmitted via a data transmission medium, such as a signal on a communications link.

[0050] From the foregoing, it will be appreciated that although specific embodiment of technology have been described herein for purposes of illustration, various modifications may be made without deviating from the spirit and scope of the invention. For example, the class definitions that have been described using XML schema can be equivalently described using other class definition tools such as a C class. The classes described can be instantiated in memory and be initialized with information. Therefore, while the invention has been described in terms of several embodiments, those skilled in the art will recognize that the invention is not limited to the embodiments described, but can be practiced with modification and alteration

within the spirit and scope of the appended claims. The description is thus to be regarded as illustrative instead of limiting.